A Common Optimization Framework for Multi-Robot Exploration and Coverage in 3D Environments

Alessandro Renzaglia, Jilles Dibangoye, Vincent Le Doze, Olivier Simonin

A Common Optimization Framework for Multi-Robot Exploration and Coverage in 3D Environments

Alessandro Renzaglia[1,2], Jilles Dibangoye[2], Vincent Le Doze[2] and Olivier Simonin[2]

Abstract This paper studies the problems of static coverage and autonomous exploration of unknown three-dimensional environments with a team of cooperating aerial vehicles. Although these tasks are usually considered separately in the literature, we propose a common framework where both problems are formulated as the maximization of online acquired information via the definition of single-robot optimization functions, which differs only slightly in the two cases to take into account the static and dynamic nature of coverage and exploration respectively. A common derivative-free approach based on a stochastic approximation of these functions and their successive optimization is proposed, resulting in a fast and decentralized solution. The locality of this methodology limits however this solution to have local optimality guarantees and specific additional layers are proposed for the two problems to improve the final performance. Specifically, a Voronoi-based initialization step is added for the coverage problem and a combination with a frontier-based approach is proposed for the exploration case. The resulting algorithms are finally tested in simulations and compared with possible alternatives.

Keywords Multi-Robot Systems · Cooperative Exploration · Optimal Coverage

1 Introduction

Multi-robot teams, especially when involving aerial vehicles, are extremely efficient systems to help humans in acquiring information on large and complex areas [30, 10]. In these scenarios, two fundamental tasks are static coverage and exploration. In the coverage problem, the robots have to find the static set of positions that optimizes a certain coverage criterion, e.g. the portion of the

[1] Univ. Grenoble Alpes, Inria, 38000, Grenoble, France · [2] INSA Lyon, CITI Lab, Inria, Chroma, Lyon, France
(e-mail: alessandro.renzaglia@inria.fr, jilles.dibangoye@inria.fr, vincent.le-doze@inria.fr, olivier.simonin@inria.fr)

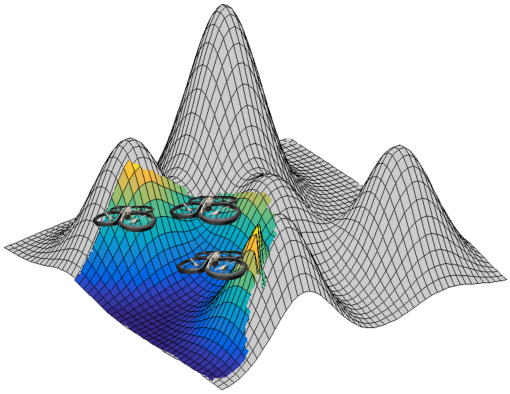

Fig. 1 Mission goal: a team of cooperating UAVs has to gather information on an unknown 3D environment (gray surface), guided by the observations collected online (blue region). Depending on the mission, the final goal is either to explore the environment or to statically cover it.

monitored environment. In the exploration, the objective is to generate the paths that allow the robots to observe the entire environment in a minimum time. In both cases, the region of interest is assumed to be unknown and the robots have to retrieve the necessary information during the mission (Fig. 1). For their relevance, these problems have received wide attention in the robotics community and numerous solutions have been proposed in the last years. However, they have been usually treated separately, proposing different formulations and approaches, and not as particular cases of a more general information-based problem. In both cases, the robots are indeed called to navigate through an unknown environment and cooperate to maximize the observed area.

The main contributions of this paper are: to present a common framework to formulate visual coverage and exploration, and to design two variants of a decentralized optimization-based solution for the two considered problems. The principal advantage of formulating both problems in such a unified framework is the transfer of knowledge, e.g., theories, methods and insights can be more easily extended from one setting to the other. In this paper, we make use of an observation-based optimization problem solved employing local stochastic approximation of a suitable optimization function which was initially introduced for the coverage problem as a building block of our solution for the exploration task. More specifically, the two problems are presented as optimization problems where, based on collected data, the robots try to online maximize the portion of observed environment. To do so, we define single-robot optimization functions whose maximization leads to a decentralized solution of the considered problems. A common optimization framework based on stochastic approximations, exploiting the data acquired during the task and exchanged among neighbors, is proposed to efficiently obtain a solution and respect the constraints of the problem. However, as any local approach, if

no prior information on the environment is exploited, the solution suffers from local optima limitations. To overcome this problem, we propose two specific solutions for the coverage and exploration problems.

Local minima in a coverage task make the team converge to configurations that can be significantly far from the global optimum of the problem. These solutions are strongly dependant on the initial state of the team and a way to overcome this drawback can be found in initializing the optimization with a favorable configuration. To this end, the presence of a partial knowledge on the environment can be an important source of information to exploit. This kind of information is available in many real situations, where a coarse knowledge of the shape, or of few specific features, is available or can be gathered before the mission starts. Based on this information, we propose a Voronoi-based partition of an approximated version of the environment to cover to obtain an initial configuration from which the local online optimization can start. On the other hand, in the exploration case, the optimization can be locally stuck and the robots would need external global inputs to identify new directions to pursue and be sure not to leave any part of the environment unexplored. To this end, we present a solution that combines the stochastic optimization with a frontier-based approach able to provide to each robot this global information whenever required to ensure the completeness of the exploration. This solution has been recently presented in [28].

The rest of the paper is organized as follows. The next section reviews the related work for both the considered problems. Section 3 presents the common formulation and the definition of the two optimization functions. In Section 4, the stochastic approximation based optimization method adopted in this work is presented. The two specific solutions to overcome the limitations due to the locality of the proposed approach are then described in Section 5. The resulting algorithms are finally tested in simulation and the results for the coverage and exploration problems are provided in Sections 6 and 7 respectively.

2 Related Work

In this section, we provide an overview of the most relevant related literature on static coverage and autonomous exploration, with a particular focus on the works taking into account three-dimensional environments and involving aerial vehicles. Since these two problems have been usually studied separately in the literature, we here present two distinct discussions on the respective related works and their connections with our contribution.

2.1 Static Coverage

The optimal coverage of a given area with a team of mobile sensors is a widely studied problem in literature and several formulations have been considered over the years. [13] presented a first classification of coverage problems, including three varieties: blanket, barrier and sweep coverage. The version considered

in our work corresponds to the blanket coverage, defined as finding the static configuration which optimizes a given coverage criterion. A fundamental work on this problem was presented by Cortés *et al.* in [7], where an approach based on the centroidal Voronoi partition was presented to drive the agents, moving in two-dimensional convex areas, toward the final configuration. Successively, numerous papers proposed extended versions of this solution, taking into account more complex environments and sensing models, but still making the assumption of perfect initial knowledge on the environment. [14] studied the case of a coverage criterion based on the visibility of the environment, assumed to be orthogonal, and using omnidirectional sensors with unlimited range. More recent works constantly increased the complexity of the sensing models and start studying the scenario of sensors moving in three dimensions. In [27], the authors presented a solution for the optimal visual coverage of a planar surface with a team of aerial vehicles equipped with down-facing cameras. The solution is based on a distributed control law maximizing a coverage-quality criterion, where the quality is considered decreasing with the altitude. A similar scenario is considered in [12], but with the objective of not leaving any area unobserved in between the fields of view of the visual sensors mounted on the UAVs, while penalizing overlaps between them. Still considering planar surfaces, [5] studies a more complex model for the UAVs sensors, which are here Pan-Tilt-Zoom (PTZ) cameras, leading to a more complex planning problem. Passing to 3D surfaces, the case of visual optimal coverage with a network of omnidirectional sensors deployed on the surface is tackled in [33], where the authors present a solution based on Cat Swarm Optimization. A similar problem but using UAVs is instead studied in [34], where the solution is again based on the concept of Voronoi tessellation of the environment. Contrary to our case, in this work the environment to cover is assumed to be initially known. [39] proposed an alternative approach, based on recursive convex optimization, for multi-camera deployment to visually cover a three-dimensional object, which is still considered perfectly known from the beginning.

Most of these solutions, exploiting geometric and gradient based approaches, often require strong assumptions on the environment or on the constraints of the problem. To deal with more challenging scenarios, derivative-free approaches can be suitable alternatives, as shown in [29] where a centralized, stochastic optimization algorithm was adopted to drive a team of UAVs to cover unknown 3D terrains. A modified version of this algorithm, proposing sparse regression techniques, was then presented in [35] where a target detection problem is formalized as a coverage problem with the objective of finding minimum-time trajectories. The main limitation of this class of solutions remains the possibility of having a strong dependency on the initial conditions and their effect on the final solution. Our contribution on this problem aims to significantly reduce this problem and so overcome this important limit.

2.2 Autonomous Exploration

The autonomous exploration is a fundamental task in multi-robot applications and for this reason it has been largely studied and presents a very rich existing literature. [20] provides an extensive overview and comparison of some of the main solutions. Another interesting survey, more focused on the communication aspects, can then be found in [1].

Most of the existing strategies for autonomous exploration exploit the frontiers between the already explored area and the still unknown environment to guide the robots during the mission. The first frontier-based solution, for a single robot, was presented by Yamauchi in his seminal paper [37], and then extended to the multi-robot case in [38]. In this work, each robot simply selected its closest frontier point at each iteration until the environment is completely explored. Following these first solutions, numerous variations, especially considering the exploration of two-dimensional areas, have been proposed in the following years. An expensive but efficient centralized approach to solve the problem of simultaneously considering traveling costs and gain utilities was presented [6]. [3] proposed a decentralized approach that allocates frontier points based on a rank, in terms of travel distance, to obtain a well balanced spatial distribution of robots in the environment. The problem of frontiers assignment in multi-robot exploration is also studied in [11], where the authors present a traveling salesman problem formulation considering all current possible goals for each robot. Even though this solution can provide shorter exploration times with respect to alternative solutions, it implies a very high computational cost. Besides the frontier allocation problem, another crucial aspect of multi-robot applications can be to comply with communication requirements. In [2], the authors consider this problem by presenting a strategy that allows agents to coordinate in order to form a network that satisfies recurrent connectivity constraints.

Considering 3D scenarios, a direct extension of a frontier-based method has been presented in [8]. In three dimensions, even the simple process of identifying the frontiers during the exploration becomes more challenging and costly and has been a subject of research [40]. In [32], the authors propose an analysis of the traversability of the scanned area and an improved K-means algorithms to reduce the dimension of frontiers in the map and improve the efficiency of the results but, contrary to our case, the exploration is carried out by ground robots. A solution, also based on frontiers, to guide a team of UAVs with embedded vision in a three-dimensional environment is then presented in [24]. The novelty of this work is to propose the exchange of only frontier points between robots instead of the entire grid map to reduce communication bandwidth.

In complex three-dimensional environments, constantly relying on frontiers has however some important disadvantages. The computation cost for efficient frontier allocation algorithms becomes very high and selecting the optimal observation point for a given frontier is often not trivial. To overcome these problems, the solution that we propose for this case has the intent

to significantly limit the use of frontiers thanks to the combination with the decentralized local optimization algorithm.

3 A Common Problem Formulation for Coverage and Exploration

The goal of the multi-robot planning problem considered in this paper is to autonomously acquiring information of an unknown surface $\mathcal{S}$ in a 3D environment with a team of UAVs. This problem can be formally defined upon the tuple $(N, \mathcal{S}, \mathcal{P}, \mathcal{C})$ where:

- N is the number of UAVs $r \in \{1, \dots, N\}$;
- $\mathcal{S}$ is the unknown set of 3D points (voxels), denoted as $s \in \mathcal{S}$, representing the surface of interest;
- $\mathcal{P}$ is the set of all robots' positions up to the current time T, $\{\mathbf{P}_t^{(r)}\}_{t=1,\dots,T}^{r=1,\dots,N}$, with $\mathbf{P}_t^{(r)} \doteq [x_{1,t}^{(r)}, x_{2,t}^{(r)}, x_{3,t}^{(r)}]$ being the position of the UAV r at time step t;
- $\mathcal{C}(\mathbf{P}_t^{(r)}) \preceq 0$ is the set of constraints that each UAV position $\mathbf{P}_t^{(r)}$ must satisfy, e.g. robot-to-obstacle and robot-to-robot collision avoidance, minimum and maximum height of flight.

Let us also define the portion of the surface $\mathcal{S}$ observed at time t by the robot r as $O_t^{(r)}(\mathcal{S})$. A point s is considered as observed by a UAV if: 1) the point and the robot are connected by a line-of-sight and 2) they are at a distance smaller than the sensing range. Since the environment is initially unknown, this information can be uniquely retrieved on-line based on the acquired measurements.

Optimal static coverage and exploration are two very related problems that can be formulated by defining two slightly different optimization functions, J_C and J_E respectively, both in terms of this tuple $(N, \mathcal{S}, \mathcal{P}, \mathcal{C})$. The goal of these optimization problems is then to find the UAVs' trajectories $\{\mathbf{P}_t^{(r)}\}_{t=1,\dots,T}^{r=1,\dots,N}$ that, by safely navigating through the three-dimensional environment, maximize the acquired information and lead the team to *i)* either find the static configuration that maximizes the surface visibility in the coverage case or *ii)* minimize the time in which the entire surface is observed for exploration.

Formally, the coverage problem can be defined by introducing the optimization function $J_C^{(r)}$ as follows:

$$J_C^{(r)}(t) = \sum_{s \in O_t^{(r)}} \frac{1}{|\mu_t(s)|} \tag{1}$$

where $\mu_t(s)$ is the subset of UAVs that can observe the point s at time t, i.e.

$$\mu_t(s) = \{r \in \{1, \dots, N\} \mid s \in O_t^{(r)}(\mathcal{S})\}. \tag{2}$$

In other words, the terms $|\mu_t(s)|$ has the role of penalizing redundant coverage by counting how many robots are observing the same element s at the same time, equally distributing the rewards among them and, as a result, forcing

the team to spread out over the environment. Note that the objective function (1) does not depend on the past states, but only on the robots configuration at a given time t.

In the exploration problem, being the environment completely unknown, the optimal solution which minimizes the mission time cannot be found off-line. The exploration needs instead to be guided by suitable heuristics based on the information on the environment acquired on-line during the mission. To this end, a local optimization function similar to (1) can be defined for each robot r to maximize the portion of observed environment at each time step. Before to introduce this function, let us firstly define the surface $E_t(\mathcal{S})$ explored up to time t as a function of the observations $O_k^{(r)}$ collected at every time k by each robot r:

$$E_t(\mathcal{S}) = \bigcup_{r=1}^{N} \bigcup_{k=1}^{t} O_k^{(r)}(\mathcal{S}) \tag{3}$$

and, for each voxel s, the set of robots $e(s)$ that observed it for the first time:

$$e(s) = \{r \in \{1, \ldots, N\} \mid s \in O_{t_s^*}^{(r)}(\mathcal{S}), s \in E_{t_s^*}(\mathcal{S}) \backslash E_{t_s^*-1}(\mathcal{S})\}, \tag{4}$$

with t_s^* being the time of first observation of the voxel s. We can so define the objective function for the robot r as follows:

$$J_E^{(r)}(t) = \sum_{s \in E_t(\mathcal{S})} \frac{\delta_{e(s)}^r}{|e(s)|} \quad \text{where } \delta_{e(s)}^r = \begin{cases} 1 & \text{if } r \in e(s) \\ 0 & \text{if } r \notin e(s) \end{cases} \tag{5}$$

The difference with (1) is that now, by maximizing this function, the robots try at each iteration to maximize the quantity of *newly* observed points s (δ term), while minimizing the redundant observations (i.e. simultaneous exploration of the same areas by more than one robot). This is due to the term $|e_t(s)|$, analogous to the term $|\mu_t(s)|$ in (1). Note that the sum over all the robots of the functions $J_E^{(r)}(t)$ is simply the total explored environment up to time t, i.e.,

$$\sum_{r=1}^{N} J_E^{(r)}(t) = \sum_{r=1}^{N} \sum_{s \in E_t(\mathcal{S})} \frac{\delta_{e(s)}^r}{|e(s)|} = |E_t(\mathcal{S})|. \tag{6}$$

Once these objective functions have been defined, our scope is to show that the optimization problems can be approached with the same kind of stochastic optimization algorithm. Additionally, we show that, being the optimization functions a combination of N single-robot functions, the solution can be completely decentralized and each robot can decide its next step only based on the information in its possession.

4 Local Stochastic Approximation-based Optimization

A first problem to solve is that the optimization functions J_C and J_E do not have a known analytical expression because of their dependency on the unknown environment to observe and, as a result, they cannot be directly optimized. However, each robot can obtain their values at a given time t, based on the data collected on-line and the information shared with the other teammates. By exploiting these data, these functions can be approximated and the result used to tackle the optimization problem [31]. To this end, we consider a decentralized version of a recent stochastic optimization algorithm, the Cognitive-based Adaptive Optimization (CAO) algorithm. This algorithm was originally developed by Kosmatopoulos in [22] and has been then proposed in various, mostly centralized, versions to tackle multi-robot planning problems, such as: multi-robot deployment [29], [30], multi-robot localization and mapping [21] and target detection [35] .

The algorithm is composed of two main steps: *i)* an approximation function is computed at each iteration to have a local estimation of the objective function J; *ii)* the obtained local approximation is maximized, respecting the problem constraints, to generate the next robot position. These two steps are detailed below.

4.1 Objective Function Approximation

At each time t, J is approximated according to:

$$\hat{J}_t^{(r)}\left(x_{1,t}^{(r)}, x_{2,t}^{(r)}, x_{3,t}^{(r)}\right) = \vartheta_t^T \, \phi\left(x_{1,t}^{(r)}, x_{2,t}^{(r)}, x_{3,t}^{(r)}\right) \tag{7}$$

where ϑ_t is a vector of parameter to estimate and ϕ the nonlinear vector of regression functions. In this work, we fix these functions in ϕ to be polynomials of the state variables up to the third-grade, i.e. the vector components ϕ_i are all the possible combinations given by:

$$\left(x_1^{(r)}\right)^u \left(x_2^{(r)}\right)^v \left(x_3^{(r)}\right)^z \quad \text{s.t.} \quad u, v, z \in \{0, 1, 2, 3\}, \;\; u + v + z \leq 3\,. \tag{8}$$

The choice of the basis functions is not unique and other solutions can also be adopted. However, it is worth remarking that this approximation is strictly local and it has no ambition to reproduce the original function in the entire exploration space. Its role is exclusively to provide an estimation of J to allow each robot to select its next position, so it needs to be reliable only in the proximity of the current robot's position. The parameter estimation vector ϑ_t, which has dimension $L = 20$, is calculated at each time t based on a limited set of past measurements according to:

$$\vartheta_t = \underset{\vartheta}{\operatorname{argmin}} \frac{1}{2} \sum_{\ell=\ell_t}^{t-1} \left(J_\ell^{(r)} - \vartheta^T \phi\left(\mathbf{P}_\ell^{(r)}\right)\right)^2 \tag{9}$$

where $\ell_t = \max\{0, t - L - T_h\}$ with T_h being a user-defined non-negative integer and $J_\ell^{(r)}$ is the value of the objective function for the robot r at time ℓ based on the collected data. The problem in (9) is an ordinary - unweighted - least square problem linear in the parameters θ_i. Its solution can be for instance obtained by considering the Moore-Penrose inverse of the matrix $X_{\ell i} = \phi_i\left(\mathbf{P}_\ell^{(r)}\right)$, where ϕ_i are the components of the vector ϕ defined in (8). Note that using only a limited set of past values has the advantage to significantly reduce the computation time required to obtain the parameter vector ϑ_t but the resulting function is a reliable approximation only locally and cannot be used for global optimization.

4.2 New State Selection

Once the approximation $\hat{J}_t$ is built according to (7) and (9), the next robot position is obtained performing a local random search. A set of admissible new positions, i.e. which respect the problem constraints, is generated by randomly perturbing the current state and tested on $\hat{J}_t$. More formally, the set of M possible states is constructed as follows:

$$\mathbf{P}_t^{(r,m)} = \mathbf{P}_t^{(r)} + \alpha_t\, \zeta_t^{(r,m)}, \forall m \in \{1, \ldots, M\}\,, \tag{10}$$

where $\zeta_t^{(r,m)}$ is a zero-mean, unit random vector with dimension equal to the dimension of $\mathbf{P}_t^{(r)}$ and α_t represents the exploration step. The standard condition that α_t must satisfy to guarantee the convergence to a local optimum (see e.g. [4]) is:

$$\lim_{t\to\infty} \alpha_t = 0, \quad \sum_{t=1}^{\infty} \alpha_t = \infty, \quad \sum_{t=1}^{\infty} \alpha_t^2 < \infty\,. \tag{11}$$

It is worth noting that this conditions must be satisfied only for the coverage problem, where the robots need to converge to a local optimum of the problem. On the contrary, in the exploration case, we do not have the necessity of a time-dependent optimization step since there is no interest in moving less than the maximum feasible step and there is no convergence until the total environment is explored.

The candidates that correspond to non-feasible positions, i.e. that violate the constraints $\mathcal{C}$, are neglected and the new robot position is $\mathbf{P}_{t+1}^{(r)} = \mathbf{P}_t^{(r,m^*)}$, where m^* is:

$$m^* = \underset{m\in\{1,\ldots,M\}}{\operatorname{argmax}}\ \hat{J}_t^{(r)}\left(\mathbf{P}_t^{(r,m)}\right)\,. \tag{12}$$

The random generation of the candidates is crucial for the efficiency of the algorithm because it ensures that the space is correctly sampled, without any persistent bias. In other words, it allows gathering the necessary information to reconstruct an unbiased approximation function and so guarantees that $\hat{J}_t$ is a reliable estimate of the unknown function J (see [22], [17] for more

details). We finally remark that in this work we focus only on high-level planning, without explicitly considering any dynamics constraints. However, more restricting assumptions, including the robot's dynamics, and so reducing the space of feasible positions would affect only the definition of $\mathcal{C}$ but not the algorithm.

5 Overcoming Local Optima

The optimization algorithm presented in the previous section can be employed by each robot to maximize the information gathered on-line on the environment and, depending on the choice of the optimization function, achieve the coverage or exploration task. However, in both cases, the proposed approach has the drawback to be strictly local and be subject to the limitations due to local optima. To overcome this problem, we propose two different strategies for the coverage and exploration tasks, allowing the team to respectively finding a better static configuration or having the guarantees of completing the exploration of the environment.

5.1 Static Coverage

In the coverage problem, the multi-robot team has to find the best static configuration which allows a maximum observation over the environment. If, as in our case, this search is exclusively conducted online, without the help of global information, the system can converge to unsatisfactory sub-optimal configurations. In this scenario, a critical choice that significantly influences the final result is the initial deployment from which the robots start their optimization. In many practical situations, some prior information on the environment to cover, even if very limited and uncertain, can be exploited to compute initial suitable configurations.

In this section, we intend to study the effect of such an initialization phase and, in particular, we propose the use of the constrained Centroidal Voronoi Tessellation [26], computed over an approximating surface, to achieve this initial step. The optimization process is thus modified by the addition of a new layer to exploit the possible presence of partial prior information. The resulting method, exploiting the complementarity between stochastic and geometric optimization, significantly improves the performance of the previous approach by reducing the probability of a far-to-optimal final solution. Furthermore, the number of iterations needed by the online algorithm to converge, and so the time required to achieve the mission, is also reduced.

To obtain the initial deployment, we assume that few real or predicted points of the surface to cover are available. Based on this information, standard methods for surface reconstruction can be applied to build an approximating

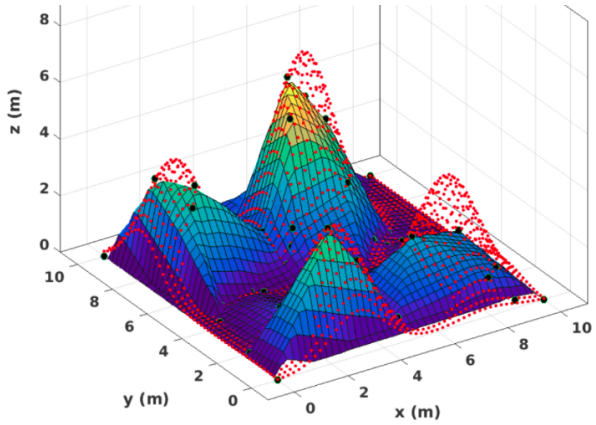

Fig. 2 Surface approximation: the real surface (red points) is approximated by a piecewise linear approximation based on 50 randomly selected points (black points).

surface $\mathcal{S}'$[1]. To obtain the surface $\mathcal{S}'$, we propose to use a piecewise linear interpolation of the available points $\{\mathbf{q}_i^*\}_{i=1}^n$, as shown in Fig. 2.

The next step is to compute a partition on this approximating surface that allows us to have a balanced distribution of the UAVs over the environment and define their initial positions. To do so, we propose to exploit the centroidal Voronoi tessellation.

Voronoi tessellation is a fundamental concept in Locational Optimization theory [26]. Given a bounded set $\Omega \in \mathbb{R}^n$ and N points $\{\mathbf{z}_i\}_{i=1}^N$, the Voronoi tessellation of Ω generated by $\{\mathbf{z}_i\}_{i=1}^N$ is $\{V_i\}_{i=1}^N$, where:

$$V_i = \{\mathbf{u} \in \Omega : |\mathbf{u} - \mathbf{z_i}| < |\mathbf{u} - \mathbf{z_j}| \; \forall j \neq i\}, \; i = 1, \ldots, N \, . \tag{13}$$

The tessellation is then called Centroidal Voronoi Tessellation (CVT) if $\mathbf{z}_i = \mathbf{z}_i^*, \forall i = 1, \ldots, N$, where $\mathbf{z}_i^*$ is the centroid of the i-th region, defined as:

$$\mathbf{z}_i^* = \frac{\int_{V_i} \mathbf{u} \, d\mathbf{u}}{\int_{V_i} d\mathbf{u}} \quad \text{for } i = 1, \ldots, N \, , \tag{14}$$

A commonly adopted solution to converge to a CVT, starting from an arbitrary set of generators, is the Lloyd's algorithm [23], [7]. This algorithm is very efficient for simply planar regions but not particularly suitable for complex scenarios since it requires the computation of the Voronoi partition corresponding to the set of points at every iteration. This step is often too difficult for arbitrary surfaces in 3D and probabilistic algorithms are more efficient. In this paper, we adopt a modified version of the MacQueen's algorithm, firstly proposed in [19] and then extended to general surfaces in [9]. Each iteration of this algorithm is composed of the following steps:

[1] As we show in Section 6, even without an informative prior it is still possible to generate an initialization able to improve the final result.

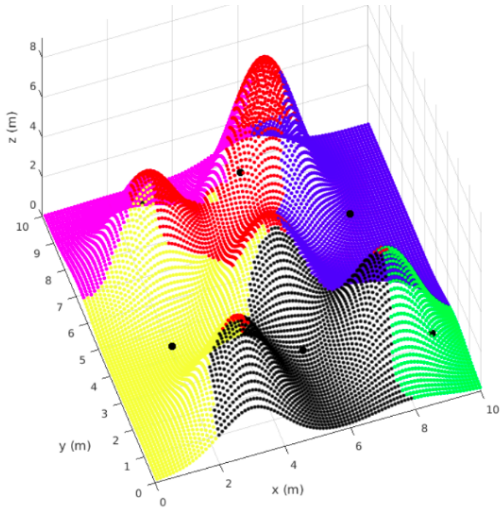

Fig. 3 Constrained CVT with six generator points.

1. Fix a positive integer q and the constants $\{\beta_i, \gamma_i\}_{i=1,2}$ such that: $\beta_2 > 0$, $\gamma_2 > 0$, $\beta_1 + \beta_2 = \gamma_1 + \gamma_2 = 1$;
2. Choose a set of N random points $\{\mathbf{z}_{i=1}^N\} \in \mathcal{S}'$ and set $j_i = 1$ for $i = 1, \ldots, N$;
3. Select randomly q samples $\{\mathbf{y}_r\}_{r=1}^q \in \mathcal{S}'$;
4. For $r = 1, \ldots, q$ define $\mathbf{z}_{i_r^*}$ as the closest $\{\mathbf{z}\}_{i=1}^N$ to $\mathbf{y}_r$;
5. For $i = 1, \ldots, N$ define the set W_i as the set of all samplings $\mathbf{y}_r$ closest to $\mathbf{z}_i$, compute the average $\mathbf{y}_i^*$ of the set W_i and define:

$$\hat{\mathbf{z}}_i^* \leftarrow \frac{(\beta_1 j_i + \gamma_1)\mathbf{z}_i + (\beta_2 j_i + \gamma_2)\mathbf{y}_i^*}{j_i + 1}, j_i \leftarrow j_i + 1; \tag{15}$$

a projection operator is then used to impose the surface constraint, i.e. $\mathbf{z}_i^* = proj(\hat{\mathbf{z}}_i^*)$, and the set of $\{\mathbf{z}_i^*\}$ becomes the new set $\{\mathbf{z}_i\}_{i=1}^N$.

These steps are repeated until a convergence criterion is fulfilled. An example of a CVT of a non-planar surface is shown in Fig. 3. It is worth noticing that such a configuration does not provide any guarantee in terms of visibility of the environment and so it does not represent an optimal solution to the visual coverage problem. Its role is exclusively to generate a first deployment from which the optimization algorithm will start. An alternative approach could be to find the global optimum solution corresponding to the approximated surface. However, we do not consider this possibility for two reasons: firstly, this initialization step can be also performed online as part of a bigger mission, and the computational cost and time required to solve the global optimization problem would be excessive for the limited resources usually available in these scenarios; secondly, this solution would still be a sub-optimal solution of the original problem because computed over a coarse approximation of the real surface.

In Section 6, the performance of the proposed initialization process will be evaluated in simulation and compared with other possible solutions to show the effectiveness of this additional layer in achieving coverage tasks.

5.2 Exploration

In the exploration task, a solution only based on a local optimization approach cannot guarantee to lead the robots to explore the entire environment. In other words, as long as new regions are explored as the result of a robot trajectory, the algorithm can provide a direction to follow, but when a robot remains completely surrounded by previously explored regions, the optimization algorithm only generates random movements and needs external input to identify new unexplored areas. To overcome this problem, we propose to combine the stochastic optimization solution with a frontier-based approach. In practice, whenever the gain of information ceases to increase, the set of remaining frontiers in the environment is considered and the robot is driven towards the closest one. This additional step allows the robot to efficiently overcome the deadlocks and restart the local optimization from the new position. The resulting decentralized algorithm, called the Frontier-based Cognitive Adaptive Optimization (FCAO) algorithm, has the goal to overcome the limitations that these two approaches have separately while maintaining their strengths. In particular, alternating local explorations with frontier assignments leads to guarantee the completeness of the exploration as in any frontier-based method. The structure of the FCAO is reported in Algorithm 1, where the function CAO is called until the local optimization is no more able to improve the objective function J_E. In this case, the function Closest Frontier selects a new goal among the list of available frontiers. The exploration is completed when this list is empty.

To build the list of frontiers, each robot makes use of its own global map that is updated by merging the collected data with local versions of the maps received from other robots (see [38] for a more thorough description of this architecture). The fundamental advantage of this strategy is that, even though some information needs to be shared, the decision is completely decentralized. This makes the system robust with respect to single robots' failures because, if a robot stops transmitting information, the other teammates can continue the mission until the entire environment is explored. Additionally, being the system asynchronous, robots do not need to wait for other robots to take their decisions.

The limitation of this method for frontiers allocation is the possibility of redundancy in the exploration, i.e. the same frontier may be assigned to more than one robot. This drawback is also present in our approach but it is limited by the fact that frontiers are used only rarely and asynchronously by the robots.

Algorithm 1: The FCAO Algorithm

```
function FCAO(w, α, E)
    Initialize t, δt = [t − w : t] and P_δt.
    repeat
        CAO(t, P_δt, E, α)
        P_{t+1} ← ClosestFrontier(P_t, E)
        t ← t + 1
    until P_{t+1} = P_t;

function CAO(t, P_δt, E, α)
    repeat
        Update J_E with new observations
        Update ϑ_t according to Equation (9)
        Generate points (ζ^(m))_{m∈{1,...,M}} randomly
        Select m* according to Equation (12)
        P_{t+1} ← P_t + αζ^(m*)
        t ← t + 1
    until J_E(t) = J_E(t − 1);

function ClosestFrontier(P_t, E)
    F = CandidateFrontiers(E)
    return SelectClosestFrontier(F, P_t)
```

6 Performance Analysis for Coverage

The solution presented in Section 5.1 for the visual coverage has been tested and evaluated in simulation. Our primary objective is to show the importance of an initial positioning as a way to overcome the limits of online local optimization. To this end, after testing this approach with generic surfaces, we provide comparative results adopting a different or none initialization. Other advantages of using the Voronoi-based initialization, such as achieving faster convergence and increasing the robustness with respect to the environment's characteristics and team capabilities, are here also discussed. Afterwards, a study on the dependency on the quantity of initial information on the surface is provided. Finally, a more realistic simulation, in terms of both UAVs model and environment to cover, has been carried out in Gazebo to show the feasibility of this strategy in realistic scenarios.

6.1 Experimental Settings

In all the results here reported, the UAVs have sensing capabilities constrained by a limited sensing distance d_{max}. The UAVs' trajectories are defined as sequences of way-points and we assume the presence of an internal controller that allows following them. The environments are assumed to be arbitrary and two distinct scenarios have been taken into account: *i*) smooth terrains generated as mixtures of Gaussians and *ii*) an outdoor environment composed of several buildings. These environments are assumed to be unknown except for a few points, which have been randomly selected on the surfaces. In both

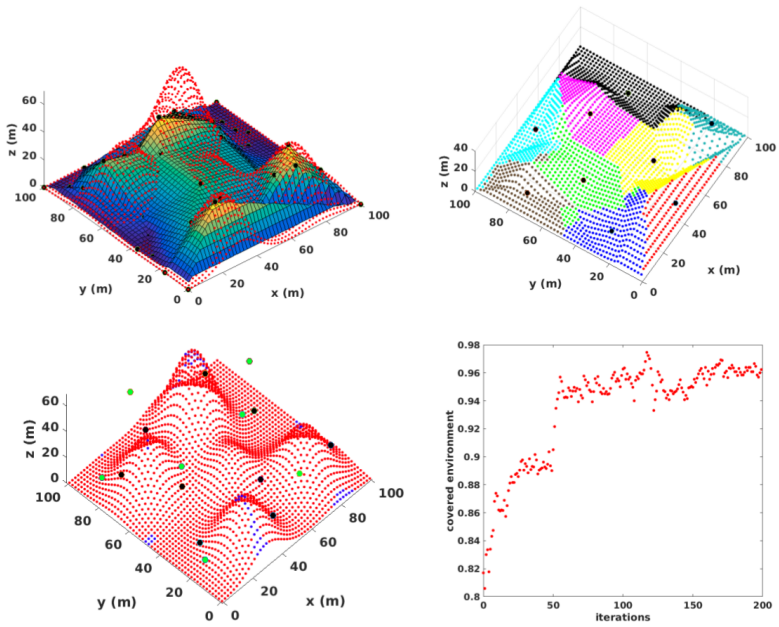

Fig. 4 Coverage obtained with 9 robots having a sensing range $d_{max} = 35m$. From top left to bottom right: 1. The real surface (in red) is approximated by a piecewise linear interpolation of few points (black dots). 2. CVT on the approximating surface. 3. Initial and final positions are in black and green respectively. The surface in red is covered by at least one UAV. 4. The portion of covered surface as a function of CAO iterations.

cases, the coverage objective function (1) has been computed by discretizing the environment and considering the number of points belonging to the surface visible from the UAVs. To facilitate the comparison between different scenarios, these objective functions have then been normalized with respect to the total surface to cover, and are so expressed as the percentage of environment covered by the UAVs. The parameters of the CAO algorithm have been set as proposed in [29].

6.2 Illustrative Example

Fig. 4 presents an illustrative example where the main steps of our approach are shown. In this scenario, 9 UAVs with a sensing range $d_{max} = 35m$ have the goal of covering an environment for which a limited set of few points is the only initial information available. The real surface is so approximated by a piecewise linear interpolation based on these points. The approximating surface is then used to compute the constrained CVT generated by the UAVs positions and the obtained configuration is the initial state from which the stochastic optimization algorithm starts. From the behavior of the coverage level during this last optimization step, it is possible to see that UAVs can

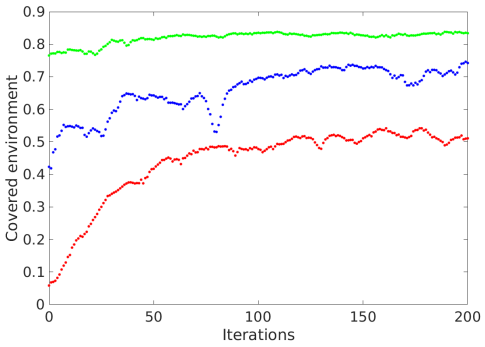

Fig. 5 Covered environment during the online optimization with a team of 8 UAVs. In red the UAVs start close one each other without initialization step, in blue their positions are initialized randomly with a uniform distribution and in green the CVT-based initialization is used as the first step of the coverage approach. The maximum coverage levels obtained by the three strategies are: 0.55, 0.75 and 0.84 respectively.

finally monitor almost the whole environment. It is important to note that the initial positions provided by the CVT can be below the real surface due to surface approximation errors. When this happens, the UAVs will converge to the closest admissible location avoiding any collision with the ground.

6.3 Comparative Results

In order to show the advantages of a CVT-based initialization, we compare the results corresponding to the coverage obtained: *i)* without any initialization, i.e. with the UAVs starting in a corner of the environment, and *ii)* with randomly initialized sets of positions generated with a uniform distribution over the region to cover.

As a first result, we analyze the coverage level achieved during the online optimization, after the three different initialization strategies, in an environment similar to the one showed in Fig. 4. From these results, reported in Fig. 5, we can see that the local optimization algorithm without any initialization is not able to find a good solution and the UAVs remain stuck in a bad local optimum. A better solution is then obtained with a random positioning of the resources. However, the CVT-based solution starts from an even higher coverage value than the final levels achieved by the two other solutions. In particular, in this case the initialization is very close to a local optimum and the online algorithm can only slightly improve the initial coverage.

To have more representative results, we carry out the same comparison but varying the number of robots and the characteristics of the environment to cover. For every fixed number of UAVs, we consider fifty different scenarios. The environments are generated as mixtures of seven Gaussians centered in random locations, with varying height and fixed variance. To generate the CVTs, a set of thirty randomly selected points is used to approximate the

	CVT initialization				**Random initialization**				**No initial.**	
N_r	Init.	σ	Max.	σ	Init.	σ	Max.	σ	Max.	σ
5	0.54	0.07	**0.67**	0.03	0.36	0.07	0.61	0.05	0.55	0.09
8	0.69	0.06	**0.83**	0.03	0.52	0.07	0.75	0.04	0.67	0.08
12	0.82	0.03	**0.92**	0.02	0.63	0.07	0.85	0.03	0.72	0.07
20	0.91	0.02	**0.98**	0.01	0.74	0.05	0.93	0.02	0.78	0.07

Table 1 Average and standard deviation of percentages of covered environment achieved over 50 scenarios, corresponding to terrains randomly generated in a 100x100m^2 square, and considering four different team sizes. The sensing range d_{max} is 25m. For the CVT and the random initialization, also the initial coverage is reported.

surface. We analyzed the performance corresponding to four different team sizes: 5, 8, 12 and 20 UAVs. All the simulations are bounded with 500 iterations, which ensure the convergence of the optimization algorithm. In Table 1, we reported for each scenario the average initial coverage (except for the non-initialized case) and the average maximum coverage achieved, all with respective standard deviations σ.

These values show that the optimization started from the Voronoi-based configuration always achieves the highest coverage level and with the smallest standard deviation. More specifically, the CVT initialization produces an improvement with respect to the random initialization comprised between 5% and 12%. Comparing then the coverage levels obtained after the initialization step, it is even more clear the advantage with respect to the random initialization. Note that these improvements also translate in a significant reduction of iterations, and so time, required to accomplish the task.

It is also interesting to analyze these performances in terms of team size. We can notice that the improvement obtained with the CVT initialization is more relevant in the intermediate cases, i.e. with 8 and 12 UAVs. This can be motivated by the fact that they represent more challenging scenarios where the optimal configuration is harder to find. In fact, with few UAVs, it is sufficient to achieve a good spread of the robots over the environment to reach configurations with non-intersecting fields of view. In such configurations, the team is close to its maximum sensing capability and so they correspond to close-to-optimal solutions. On the other side, in large teams, a high level of overlapping observations is inevitable and several distinct configurations can lead to the same total coverage of the environment (in the extreme case of a very large number of robots, even a random deployment may lead to full coverage). However, an initial dispersion of the UAVs can still be beneficial to avoid that some of them remain stuck in their initial location reducing the total sensing capability. This can be seen in the case of 20 UAVs, where the coverage without initialization is significantly lower than the other two.

6.4 Dependency on Prior Information

We now want to stress the importance of the CVT to uniformly distribute the robots over the environment and provide a good initialization even when the information on the surface to cover is extremely limited. For one scenario reported in Table 1 (12 robots, third line), we perform the same study over the fifty environments but now considering the approximating surface based on just the four points defining the boundaries of the region plus one point in the middle of the environment. The resulting approximating surface is in this case simply a pyramid. The CVT initialization for this extreme case provides an almost uniform distribution of the UAVs over the environment. The coverage level obtained after the initialization with this surface is 0.78 with a standard deviation $\sigma = 0.03$, while the final maximum coverage is 0.91 with a standard deviation of $\sigma = 0.02$. As expected, the first result shows that the initial value is lower than the one obtained with a more informative approximation, but it remains a better solution than a random initialization. Then, the online optimization is able to drive the UAVs to achieve a final coverage very close to the result of the more informative initialization. This highlights the important effect of the Voronoi tessellation to spread the team and generate a suitable initialization.

6.5 Urban Environment

To evaluate our approach in a more complex scenario, we present here the results obtained by carrying out simulations in Gazebo [15] and realized using the ROS package *tum_simulator* [36]. These simulations include more realistic models for both the environment and the system, showing the employability of our approach in realistic applications. In the proposed scenario, four simulated UAVs with a maximum visual range of $8m$ are employed to cover a $24m$x$24m$ area with several buildings. The steps illustrated in Fig. 6, from top left to bottom right, are: the initialization, computed on an extremely simplified reproduction of the environment; the final positions reached after the on-line optimization performed by the CAO algorithm; the covered environment represented as a point cloud; the coverage level achieved during the mission, here also compared with the result obtained from a different initialization (two UAVs starting close to each other on one side of the environment and the other two similarly on the opposite side). This comparison clearly shows that the initialized case begins from a higher value and converge significantly faster to the final positions, not far from the initialization.

7 Performance Analysis for Exploration

Considering now exploration tasks, the proposed FCAO algorithm has been tested in simulations and compared with possible alternative solutions. As for

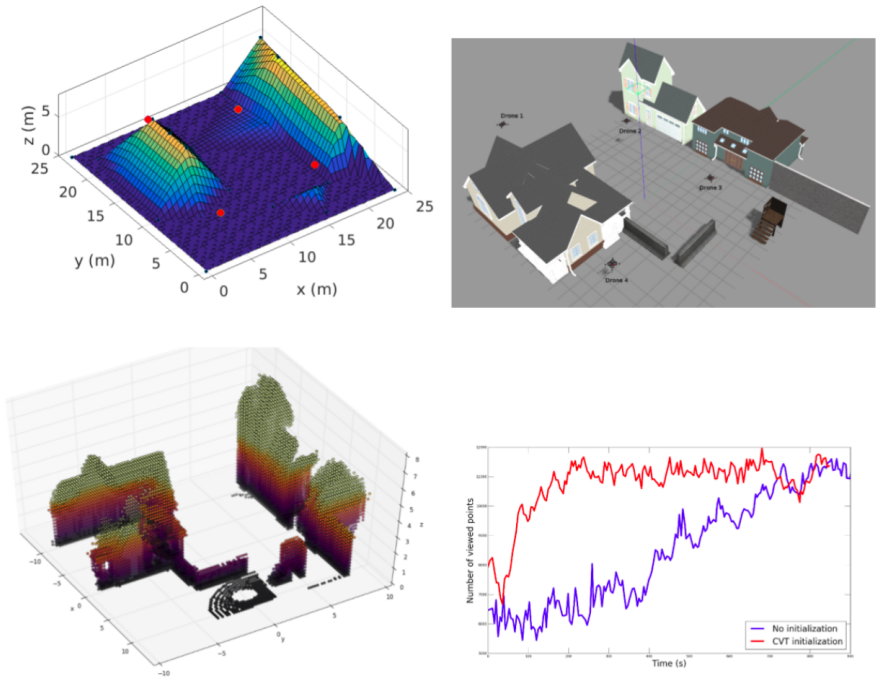

Fig. 6 Four UAVs cover a simulated environment with several buildings. From top left to bottom right: 1. The initial positions are computed exploiting an approximation of the environment. 2. These positions are optimized based on collected data to obtain the final configuration. 3. The map of the environment obtained by the team. 4. The coverage levels obtained after the CVT initialization and a non-initialized configuration.

the coverage results presented in the previous section, the surfaces to explore are arbitrary and, in a first study, created as random mixtures of Gaussians with different heights and variances. Fig. 7 presents an illustrative example with four UAVs and their respective trajectories. In all the results reported in this section, the robots' initial locations can slightly vary but they are always in a corner of the environment, close one each other.

To evaluate the performance of the proposed FCAO algorithm, it has been compared with two alternative approaches:

1. **Closest frontier:** a decentralized frontier-based algorithm, where each robot selects its closest frontier point independently from the others. This algorithm corresponds to the standard solution presented in [38].
2. **Greedy:** a centralized solution that 1) divides frontiers in clusters and considers their centroids as candidate frontiers to improve the distribution among robots and reduce redundancy in the exploration (as done for instance in [32]), and 2) assigns clusters' centroids sequentially to each robot to avoid multiple assignments of the same frontiers to different robots.

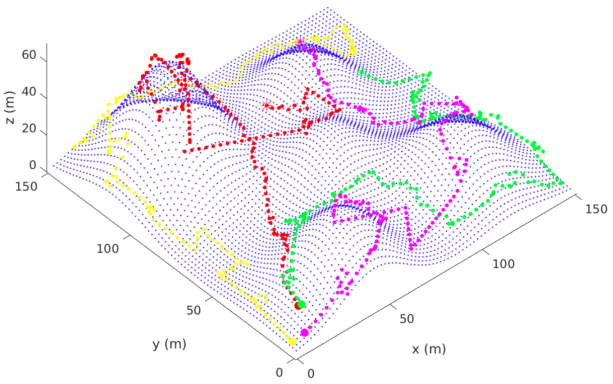

Fig. 7 Four UAVs exploring an unknown terrain.

Note that this greedy assignment algorithm is a variation of [6] considering constant frontiers utilities[2].

Fig. 8 presents a first result with eight UAVs starting from the same set of initial locations. As a first remark, it is possible to see that our solution significantly outperforms the closest-frontier algorithm. This clear effect can be explained with the high level of redundancy in the exploration only considering the closest frontiers selection. On the contrary, the local optimization efficiently spreads out the robots, avoiding important overlap in the observations. Then, it is interesting to notice that, even though the decision is decentralized, our approach obtains a performance very close to the centralized greedy solution.

Considering the same scenario, we then analyzed the computational time required to find the next robot position at each iteration. One of the strong advantages of our method is its low computational cost and it is clearly shown in Fig. 8(Right). This factor is indeed not only significantly below the time needed by the greedy algorithm but it is even faster than the simple closest frontier selection. The main computational difference here is the necessity for frontier-based algorithms to compute each time the distances between each robot and all the frontier points, an operation that becomes very expensive in three dimensions. This effect is visible even in our analysis where we do not compute the exact geodesic distance between two points, but instead we consider only the straight line connecting the two points and adjusting the altitude to pass above the surface, including a safety distance from the terrain. We can also see that the run time of the local optimizer is constant (except for the very few steps where frontiers are exploited) while the others are varying during the time because of the dependence on the number of frontiers in the map at each iteration. Moreover, being decentralized, its cost is also independent of the number of robots, contrary to the centralized greedy algorithm. The only

[2] The utilities update step as in [6] is already computationally expensive in two dimensions and for 3D environments would make the computational burden of this algorithm too high to be considered in this paper, where we focus on light solutions.

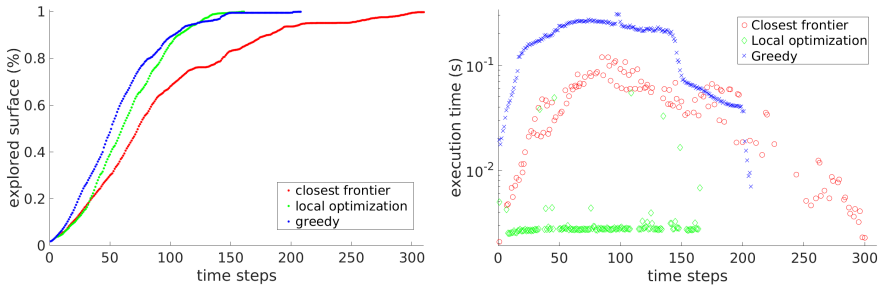

Fig. 8 The FCAO approach is compared with two alternatives: a decentralized closest frontier algorithm and a centralized greedy selection of frontiers. The results correspond to a scenario with 8 robots in an environment similar to the one shown in Fig. 7. Left: Portion of explored environment as a function of time. Right: Time required to identify the next way point at each iteration.

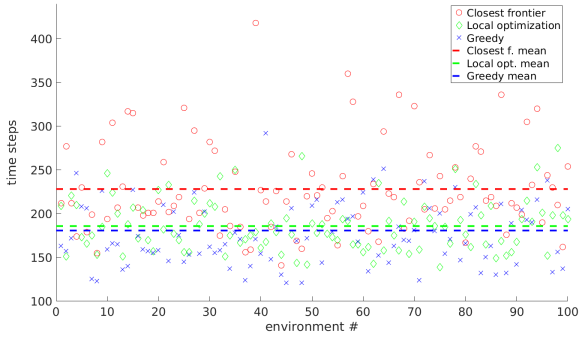

Fig. 9 Number of steps necessary with the three algorithms to complete the exploration for each of the 100 different randomly generated environments. Dashed lined represent the means for each approach.

relevant cost of the local optimization comes from solving the problem in (9), which depends only on the dimension of the vector ϕ and the number of past considered measurements, which are both constant during the exploration.

Finally, we compared these three possible solutions in 100 randomly generated environments (eight peaks with the same variance and centered in randomly selected, uniformly distributed locations). It is worth noticing that, even though these environments are relatively open, the sensing constraints compared with the peaks elevation force the robots to fly among them and not just over them, making the complexity of the environment and the task higher. Furthermore, as for the coverage case, our study is not only limited to very complex environments but also to more generic open outdoor terrains, where the redundancy problem is one of the most relevant to reduce the performance of computationally light solutions. The number of iterations needed to complete the task and the averages for each algorithm are reported in Fig. 9. Similarly to the previous results, the FCAO algorithm performs very closely to the centralized greedy algorithm, and they both significantly outperform the

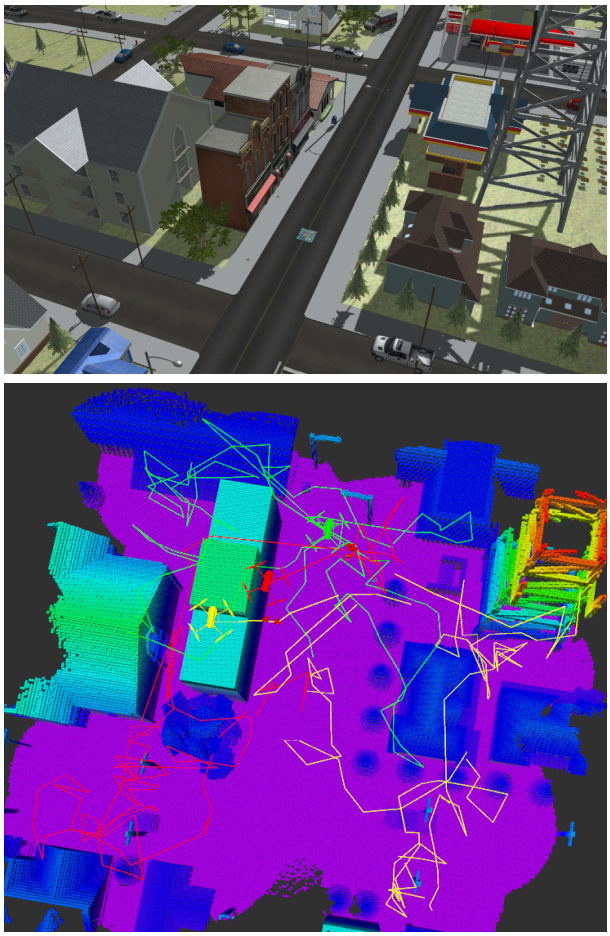

Fig. 10 Environment to explore (top) and final 3D map (bottom) obtained with three UAVs guided by the FCAO algorithm and their respective trajectories.

closest-frontier solution. We can also notice that the latter, besides showing a worse average behavior, also presents a greater variance in the results, while the other two are more robust with respect to changes in the environment.

7.1 Urban Environment

Similarly to the results of the previous section, we finally tested our approach in more complex and realistic simulations executed using ROS and Gazebo simulator. A rich urban environment including buildings, complex structures, and vehicles has been here considered. The UAV's dynamics have been here defined following the model presented in [25]. A controller, based on feedback linearization (for more details see e.g. [18]), has been implemented allowing

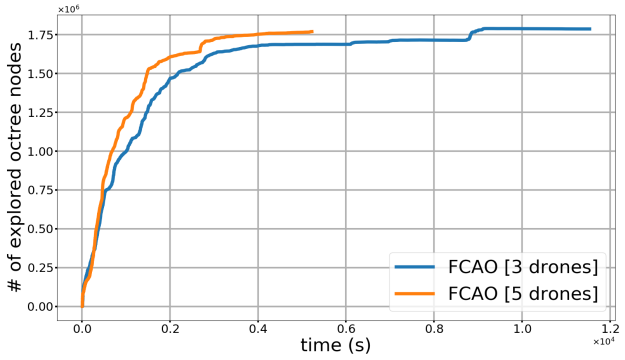

Fig. 11 Number of octree nodes explored over time for the scenario presented in Fig. 10. The result with 3 UAVs is in blue, with 5 in orange.

the translation of the new positions generated by the optimizer in rotational velocities for the rotors. The components have been developed to be as physical representative as possible, especially regarding the aerodynamics, which has been modeled after the specifics of the Intel Aero quadrotor[3]. This allows us to reproduce realistic trajectories for the drones and to show that the way-points are generated in a way that leads to feasible motion.

The UAVs obtain at each position a point cloud representation of the environment included in a down-looking, $20m$-radius half-sphere. The map of the observed environment and, if necessary, a list of frontier points are generated based on this point cloud thanks to the ROS package Octomap_server [16]. The result of the exploration conducted by three UAVs is presented in Fig. 10, where the initial environment, the final 3D map, and the UAVs trajectories are shown[4]. It is possible to see that the solution leads to a complete 3D reconstruction, i.e. without leaving any area unobserved. Furthermore, we can notice that the UAVs, after taking-off from close positions, are driven by the algorithm toward different regions to minimize the information overlap, and so resulting in a reduced exploration time. For a more quantitative analysis, in Fig. 11 we report the number of octree nodes explored over time. The result obtained in Fig. 10 is here compared to the same exploration conducted by a team of five UAVs. From this plot, we can see how the UAVs are able, in both cases, to continuously improve their knowledge of the environment, avoiding getting stuck in already explored regions. As expected, the team composed of five UAVs obtains a better result, requiring about 40% less time to complete the exploration.

[3] Our simulator is available online at `https://gitlab.inria.fr/chroma/drone-simulator`.

[4] A video of this simulation is also available at `https://team.inria.fr/chroma/files/2020/05/FCAO_2020.mp4`

8 Final Discussion

In this paper, we considered the cooperative static coverage and the autonomous exploration problems for three-dimensional environments with a team of aerial robots. The main subject of our study was the possibility to express both problems with a unique framework, showing their similarities with a structure based on the observations gathered online and trying to overcome the usual gap present in their solutions. Formally, these problems maintain a substantial difference in their nature. The solution of the coverage problem is indeed represented by the set of static positions that maximizes the total observation, while in the exploration the entire robots' trajectories are the objective of the problem. However, in our work, we showed that a common structure of single-robot objective functions can be defined and that it allows exploiting for both problems a measurement-based stochastic optimization approach to locally maximize the online acquired information. Furthermore, we showed that this approach, contrary to most of the existing solutions, leads to decentralized and light solutions, particularly suitable for computationally limited platforms.

Additional specific strategies, reflecting the still existing differences of these problems, have been then proposed in order to overcome the well-known limitations of local optimality and obtain more efficient solutions. In particular, for the coverage problem, we presented an algorithm, based on centroidal Voronoi tessellations, to generate suitable initial configurations from which the online optimization leads to significantly better final results in fewer iterations. For the exploration task, we instead proposed the FCAO algorithm which combines the local online optimization with a sparse exploitation of frontier points to obtain a light and efficient approach. In fact, this solution maintains all the advantages of the local stochastic optimization method, avoids the necessity of computing the full distance matrix between robots and frontiers at each iteration, and still guarantees the final total exploration of the environment.

The proposed methodology has been tested in simulation, considering different classes of environments and comparing the obtained results with possible alternative strategies. These results showed the efficiency of our solution for both coverage and exploration. In particular, the FCAO algorithm for cooperative exploration, despite being completely decentralized and less computationally requiring, provided results comparable with a more complex centralized approach and clearly outperformed a standard decentralized alternative.

It is worth mentioning that another significant advantage of a unified formulation for these two tasks is that it eases the process of finding new advanced solutions for problems that lie within the proposed framework, i.e. new combinations between exploration and coverage problems. Additionally, the use of a common framework, based on similar information and on the same optimization approach, could also significantly ease the execution of the two solutions in scenarios where switching between two tasks proves necessary. While the present work focused on independently solving the two problems, we plan to investigate these possible directions as part of future studies.

We also believe that both problems present further interesting research lines to improve our specific solutions. For the coverage problem, we intend to pursue the study on more informative coverage criteria, including also the quality of acquired visual data in the objective function. This will affect the generation of the initial partition since new metrics could be considered instead of standard distances for the Voronoi tessellation. For the exploration problem, we consider that relevant questions remain open on frontier-based approaches for 3D environments, such as finding fast and efficient strategies to define the best viewpoints once a frontier is identified. Finally, studying the effects of uncertainties generated by localization errors and the robustness of the proposed strategies will also be an important point to analyze.

References

1. F. Amigoni, J. Banfi, and N. Basilico. Multirobot exploration of communication-restricted environments: A survey. *IEEE Intelligent Systems*, 32(6):48–57, 2017.
2. J. Banfi, A. Q. Li, I. Rekleitis, F. Amigoni, and N. Basilico. Strategies for coordinated multirobot exploration with recurrent connectivity constraints. *Autonomous Robots*, 42(4):875–894, 2018.
3. A. Bautin, O. Simonin, and F. Charpillet. Minpos: A novel frontier allocation algorithm for multi-robot exploration. In *International conference on intelligent robotics and applications*, pages 496–508. Springer, 2012.
4. D. Bertsekas and J. Tsitsiklis. Gradient convergence in gradient methods with errors. *Journal on Optimization*, 10(3), 2000.
5. N. Bousias, S. Papatheodorou, M. Tzes, and A. Tzes. Collaborative visual area coverage using aerial agents equipped with ptz-cameras under localization uncertainty. In *18th IEEE European Control Conference (ECC)*, pages 1079–1084, 2019.
6. W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider. Coordinated multi-robot exploration. *IEEE Transactions on robotics*, 21(3):376–386, 2005.
7. J. Cortés, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.
8. C. Dornhege and A. Kleiner. A frontier-void-based approach for autonomous exploration in 3d. *Advanced Robotics*, 27(6):459–468, 2013.
9. Q. Du, M. D. Gunzburger, and L. Ju. Constrained centroidal Voronoi tessellations for surfaces. *SIAM Journal on Scientific Computing*, 24(5), 2003.
10. M. Erdelj, E. Natalizio, K. Chowdhury, and I. Akyildiz. Help from the sky: Leveraging uavs for disaster management. *IEEE Pervasive Computing*, 16(1):24–32, 2017.
11. J. Faigl, M. Kulich, and L. Přeučil. Goal assignment using distance cost in multi-robot exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3741–3746, 2012.
12. R. Funada, M. Santos, J. Yamauchi, T. Hatanaka, M. Fujita, and M. Egerstedt. Visual coverage control for teams of quadcopters via control barrier functions. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3010–3016, 2019.
13. D. W. Gage. Command control for many-robot systems. Technical report, Naval Command Control and Ocean Surveillance Center RDT and E Div San Diego, CA, 1992.
14. A. Ganguli, J. Cortés, and F. Bullo. Visibility-based multi-agent deployment in orthogonal environments. In *IEEE American Control Conference (ACC)*, 2007.
15. Gazebo robot simulator. `http://gazebosim.org`.
16. A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.
17. P. A. Ioannou and J. Sun. *Robust adaptive control*. Courier Corporation, 2012.
18. L. Jaulin. *Mobile robotics*. John Wiley & Sons, 2019.

19. L. Ju, Q. Du, and M. Gunzburger. Probabilistic methods for centroidal Voronoi tessellations and their parallel implementations. *Parallel Computing*, 28(10), 2002.
20. M. Juliá, A. Gil, and O. Reinoso. A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. *Autonomous Robots*, 33(4):427–444, 2012.
21. A. Kapoutsis, S. Chatzichristofis, L. Doitsidis, J. B. de Sousa, J. Pinto, J. Braga, and E. B. Kosmatopoulos. Real-time adaptive multi-robot exploration with application to underwater map construction. *Autonomous robots*, 40(6):987–1015, 2016.
22. E. B. Kosmatopoulos. An adaptive optimization scheme with satisfactory transient performance. *Automatica*, 45(3), 2009.
23. S. Lloyd. Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2), 1982.
24. N. Mahdoui, V. Frémont, and E. Natalizio. Cooperative frontier-based exploration strategy for multi-robot system. In *2018 13th Annual Conference on System of Systems Engineering (SoSE)*, pages 203–210. IEEE, 2018.
25. P. Martin and E. Salaün. The true role of accelerometer feedback in quadrotor control. In *IEEE International Conference on Robotics and Automation*, pages 1623–1629, 2010.
26. A. Okabe. *Spatial tessellations*. Wiley Online Library, 1992.
27. S. Papatheodorou, A. Tzes, and Y. Stergiopoulos. Collaborative visual area coverage. *Robotics and Autonomous Systems*, 92, 2017.
28. A. Renzaglia, J. S. Dibangoye, V. Le Doze, and O. Simonin. Combining stochastic optimization and frontiers for aerial multi-robot exploration of 3d terrains. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.
29. A. Renzaglia, L. Doitsidis, A. Martinelli, and E. B. Kosmatopoulos. Multi-robot three-dimensional coverage of unknown areas. *The International Journal of Robotics Research*, 31(6), 2012.
30. D. Scaramuzza et al. Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments. *IEEE Robotics & Automation Magazine*, 21(3), 2014.
31. J. C. Spall. Stochastic optimization. In *Handbook of computational statistics*, pages 173–201. Springer, 2012.
32. Y. Tang, J. Cai, M. Chen, X. Yan, and Y. Xie. An autonomous exploration algorithm using environment-robot interacted traversability analysis. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4885–4890, 2019.
33. S. Temel, N. Unaldi, and O. Kaynak. On deployment of wireless sensors on 3d terrains to maximize sensing coverage by utilizing cat swarm optimization with wavelet transform. *IEEE Trans. on Systems, Man, and Cybernetics*, 44(1):111–120, 2014.
34. M. Thanou and A. Tzes. Distributed visibility-based coverage using a swarm of UAVs in known 3D-terrains. In *IEEE International Symposium on Communications, Control and Signal Processing*, 2014.
35. K.-S. Tseng and B. Mettler. Near-optimal probabilistic search via submodularity and sparse regression. *Autonomous Robots*, 41(1), 2017.
36. TUM ROS simulator. `http://wiki.ros.org/tum\_simulator`.
37. B. Yamauchi. A frontier-based approach for autonomous exploration. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 146–151, 1997.
38. B. Yamauchi. Frontier-based exploration using multiple robots. In *Proceedings of the second international conference on Autonomous agents*, pages 47–53. ACM, 1998.
39. X. Zhang, X. Chen, J. L. Alarcon-Herrera, and Y. Fang. 3d model-based multi-camera deployment: a recursive convex optimization approach. *IEEE/ASME Transactions on Mechatronics*, 20(6):3157–3169, 2015.
40. C. Zhu, R. Ding, M. Lin, and Y. Wu. A 3d frontier-based exploration tool for mavs. In *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 348–352. IEEE, 2015.